AMENDMENTS TO THE CLAIMS

This listing of claims will replace all prior versions, and listings of claims in the application:

LISTING OF CLAIMS

1. (Currently Amended) A computer implemented process for managing exceptions throwable during execution of methods in one or more classes on a resource-constrained device, each method including protected code and an exception handler array defining exception handlers associated with the method, the process comprising:

   defining an ordering for the methods, the ordering defining the placement of the protected code in a first portion of a package; and

   combining in block form the exception handler arrays for the methods into a single exception handler table, the combining comprising positioning the exception handler arrays in a second portion of the package based on the ordering for the methods.

2. (Original) The process of claim 1 including combining all exception handler arrays for all methods in a class in the single exception handler table.

3. (Original) The process of claim 1 including combining all exception handler arrays for all methods in all classes in the single exception handler table.

4. (Original) The process of claim 1 including combining all exception handler arrays for all methods in a Java package in the single exception handler table.

5. (Original) The process of claim 1 where a method is included in a class file and the step of combining all exception handler arrays includes combining the exception handler arrays for all methods in a class file in the single exception handler table.

6. (Original) The process of claim 1 further including searching the exception handler table when an exception is thrown while executing one of the methods including locating a first matching exception in the single exception handler table.

7. (Original) The process of claim 6 where the searching step includes retrieving in order exception handler entries from the exception handler table and checking the type and range of each exception handler for the first matching exception handler.

8. (Original) The process of claim 7 further comprising stopping searching if a current exception handler does not match and is the last handler for the top most level of protected code in an associated method.

9. (Original) The process of claim 1 where the class files are Java class files.

10. (Cancelled)

11. (Cancelled)

12. (Cancelled)

13. (Previously Presented) The process of claim 1 where the resource-constrained device comprises a virtual machine configured to perform said execution of said methods.

14. (Original) The process of claim 13 where the resource constrained device is a smart card.

15. (Original) The process of claim 14 where the methods in one or more classes are grouped in a package and the package is installed on the smart card.

16. (Cancelled)

17. (Currently Amended) A method minimizing the amount of storage required for a runtime stack when executing a program, the runtime stack maintained at runtime during the execution of the program on a resource-constrained device for storing one or more frames where each frame includes a return pointer to an invoking method that called a currently executing method in the program, the method comprising: defining an ordering for methods included in the program, each method including protected code and an exception handler array defining exception handlers

associated with the method, the ordering defining the placement of the protected code in a first portion of a package;

combining in block form exception handler information for the methods included in the program into a combined exception handler table, the combining comprising positioning the exception handler arrays in a second portion of the package based on the ordering for methods included in the program; and

locating and searching the combined exception handler table when an exception is thrown during execution of one of the methods to locate the exception handler information without requiring the storage on the runtime stack of a pointer to the exception handler information.

18. (Original) The method of claim 17 where the pointer is a direct pointer to the exception handler information.

19. (Original) The method of claim 17 where the program is a Java program.

20. (Previously Presented) The method of claim 19 where the resource-constrained device comprises a virtual machine implementing a Java ™ virtual machine and configured to perform said execution of said methods.

21. (Previously Presented) The method of claim 20 where the program includes a package of methods, the methods in one or more classes, and where the virtual

machine is implemented in a resource constrained device on which the package is installed and executing.

22. (Original) The method of claim 21 where the resource constrained device is a smart card.

23. (Original) The method of claim 21 further including registering the package in a registry service at installation, the registry service maintaining a pointer and a range, the pointer indicating a location in the resource constrained device of the combined exception handler table associated with a given package, the range defining a range of addresses in the resource constrained device at which methods associated with the package are located.

24. (Original) The method of claim 23 where the step of locating includes locating a package associated with a currently executing method including comparing an address at which an exception was thrown against the range for each package registered in the registry service, the searching step including searching the combined exception handler table associated with a located package.

25. (Previously Presented) A method of converting class files into a converted applet for execution on a resource constrained device, the method comprising:

receiving one or more class files, each class file including one or more methods, each method including protected code and an exception handler array defining exception handlers catchable by the method;

defining a data structure for storing the methods and exception handlers for the converted applet including a first and second portion;

defining an ordering for the methods and loading the methods according to the ordering in the second portion of the data structure; and

combining in block form the exception handler arrays for all methods in a single exception handler table including ordering the exception handler arrays according to the ordering defined for the methods and storing the single exception handler array in the first portion of the data structure.

26. (Currently Amended) A computer implemented process for managing exceptions throwable during execution of two or more methods in one or more classes by a virtual machine on a resource-constrained device, each method included in a class and including protected code and an exception handler array defining exception handlers associated with the method, the protected code of the two or more methods stored in a first portion of a package according to an ordering, the individual exception handler arrays combined in block form and forming a single exception handler table for the two or more methods, the exception handler arrays positioned in a second portion of the package based on the ordering of the two or more methods, the process comprising:

searching the exception handler table when an exception is thrown while executing one of the methods including locating a first matching exception in the single exception handler table.

27. (Currently Amended) A computer implemented system for managing exceptions throwable during execution of methods in one or more classes on a resource-constrained device, each method including protected code and an exception handler array defining exception handlers associated with the method, the system comprising instructions to:

define an ordering for the methods, the ordering defining the placement of the protected code in a first portion of a package; and

combine in block form the exception handler arrays for all methods into a single exception handler table, the combining comprising positioning the exception handler arrays in a second portion of the package based on the ordering for the methods.

28. (Currently Amended) A computer implemented system for minimizing the amount of storage required for a runtime stack when executing a program, the runtime stack maintained at runtime during the execution of the program on a resource-constrained device for storing one or more frames where each frame includes a return pointer to an invoking method that called a currently executing method in the program, the system comprising instructions to:

define an ordering for methods included in the program, each method including protected code and an exception handler array defining exception handlers associated with the method, the ordering defining the placement of the protected code in a first portion of a package;

combine in block form the exception handler information for the methods included in the program into a combined exception handler table, the combining comprising positioning the exception handler arrays in a second portion of the package based on the ordering for methods included in the program; and

locate and search the combined exception handler table when an exception is thrown during execution of one of the methods to locate the exception handler information without requiring the storage on the runtime stack of a pointer to the exception handler information.

29. (Previously Presented) A computer implemented system for converting class files into a converted applet for execution on a resource constrained device, the system comprising instructions to:

receive one or more class files, each class file including one or more methods, each method including protected code and an exception handler array defining exception handlers catchable by the method;

define a data structure for storing the methods and exception handlers for the converted applet including a first and second portion;

define an ordering for the methods and loading the methods according to the ordering in the second portion of the data structure; and combine in block form the exception handler arrays for all methods in a single exception handler table including order the exception handler arrays according to the order defined for the methods and store the single exception handler array in the first portion of the data structure.

30. (Currently Amended) A computer implemented system for managing exceptions throwable during execution of methods in one or more classes by a virtual machine on a resource-constrained device, each method in a class described by a class file and including protected code and an exception handler array defining exception handlers associated with the method, the protected code of the two or more methods stored in a first portion of a package according to an ordering, the individual exception handler arrays combined in block form and forming a single exception handler table for two or more methods, the exception handler arrays positioned in a second portion of the package based on the ordering of the two or more methods, the system comprising instructions to:

search the exception handler table when an exception is thrown while executing one of the two or more methods including locate a first matching exception in the single exception handler table.

31. (Currently Amended) A program storage device readable by a machine, embodying a program of instructions executable by the machine to perform a process for managing exceptions throwable during execution of methods in one or more classes on a resource-constrained device, each method including protected code and an

exception handler array defining exception handlers associated with the method, the process comprising:

defining an ordering for the methods, the ordering defining the placement of the protected code in a first portion of a package; and

combining in block form the exception handler arrays for the methods into a single exception handler table, the combining comprising positioning the exception handler arrays in a second portion of the package based on the ordering for the methods.

32. (Previously Presented) The program storage device of claim 31 where the process includes combining all exception handler arrays for all methods in a class in the single exception handler table.

33. (Previously Presented) The program storage device of claim 31 where the process includes combining all exception handler arrays for all methods in all classes in the single exception handler table.

34. (Previously Presented) The program storage device of claim 31 where the process includes combining all exception handler arrays for all methods in a Java™ package in the single exception handler table.

35. (Previously Presented) The program storage device of claim 31 where a method is included in a class file and the combining all exception handler arrays includes

combining the exception handler arrays for all methods in a class file in the single exception handler table.

36. (Previously Presented) The program storage device of claim 31 where the process further includes searching the exception handler table when an exception is thrown while executing one of the methods including locating a first matching exception in the single exception handler table.

37. (Previously Presented) The program storage device of claim 36 where the searching includes retrieving in order exception handler entries from the exception handler table and checking the type and range of each exception handler for the first matching exception handler.

38. (Previously Presented) The program storage device of claim 37 where the process further comprises stopping searching if a current exception handler does not match and is the last handler for the top most level of protected code in an associated method.

39. (Previously Presented) The program storage device of claim 31 where the class files are Java™ class files.

40. (Cancelled)

41. (Cancelled)

42. (Cancelled)

43. (Previously Presented) The program storage device of claim 31 where the machine is a virtual machine implemented on a resource constrained device.

44. (Previously Presented) The program storage device of claim 43 where the resource constrained device is a smart card.

45. (Previously Presented) The program storage device of claim 44 where the methods in one or more classes are grouped in a package and the package is installed on the smart card.

46. (Cancelled)

47. (Currently Amended) A program storage device readable by a machine, embodying a program of instructions executable by the machine to perform a process for minimizing the amount of storage required for a runtime stack when executing a program, the runtime stack maintained at runtime during the execution of the program on a resource-constrained device for storing one or more frames where each frame includes a return pointer to an invoking method that called a currently executing method in the program, the process comprising:

defining an ordering for methods included in the program, each method including protected code and an exception handler array defining exception handlers associated with the method, the ordering defining the placement of the protected code in a first portion of a package;

combining in block form exception handler information for the methods included in the program into a combined exception handler table, the combining comprising positioning the exception handler arrays in a second portion of the package based on the ordering for methods included in the program; and

locating and searching the combined exception handler table when an exception is thrown during execution of one of the methods to locate the exception handler information without requiring the storage on the runtime stack of a pointer to the exception handler information.

48. (Previously Presented) The program storage device of claim 47 where the pointer is a direct pointer to the exception handler information.

49. (Previously Presented) The program storage device of claim 47 where the program is a Java™ program.

50. (Previously Presented) The program storage device of claim 49 where the machine is a virtual machine implementing a Java™ virtual machine.

51. (Previously Presented) The program storage device of claim 50 where the program includes a package of methods, the methods in one or more classes, and where the virtual machine is implemented in a resource constrained device on which the package is installed and executing.

52. (Previously Presented) The program storage device of claim 51 where the resource constrained device is a smart card.

53. (Previously Presented) The program storage device of claim 51 where the process further includes registering the package in a registry service at installation, the registry service maintaining a pointer and a range, the pointer indicating a location in the resource constrained device of the combined exception handler table associated with a given package, the range defining a range of addresses in the resource constrained device at which methods associated with the package are located.

54. (Previously Presented) The program storage device of claim 53 where the locating includes locating a package associated with a currently executing method including comparing an address at which an exception was thrown against the range for each package registered in the registry service, the searching including searching the combined exception handler table associated with a located package.

55. (Previously Presented) A program storage device readable by a machine, embodying a program of instructions executable by the machine to perform a process for

converting class files into a converted applet for execution on a resource constrained device, the process comprising:

receiving one or more class files, each class file including one or more methods, each method including protected code and an exception handler array defining exception handlers catchable by the method;

defining a data structure for storing the methods and exception handlers for the converted applet including a first and second portion;

defining an ordering for the methods and loading the methods according to the ordering in the second portion of the data structure; and

combining in block form the exception handler arrays for all methods in a single exception handler table including ordering the exception handler arrays according to the ordering defined for the methods and storing the single exception handler array in the first portion of the data structure.

56. (Currently Amended) A program storage device readable on a resource-constrained device, embodying a program of instructions executable by the machine to perform a method for managing exceptions throwable during execution of two or more methods in one or more classes by a virtual machine on a resource-constrained device, each method included in a class and including protected code and an exception handler array defining exception handlers associated with the method, the protected code of the two or more methods stored in a first portion of a package according to an ordering, the individual exception handler arrays combined in block form and forming a single exception handler table for the two or more methods, the

exception handler arrays positioned in a second portion of the package based on the ordering of the two or more methods, the process comprising:

searching the exception handler table when an exception is thrown while executing one of the methods including locating a first matching exception in the single exception handler table.

57. (Currently Amended) An apparatus for managing exceptions throwable during execution of methods in one or more classes on a resource-constrained device, each method including protected code and an exception handler array defining exception handlers associated with the method, the apparatus comprising:

means for defining an ordering for the methods, the ordering defining the placement of the protected code in a first portion of a package; and

means for combining in block form the exception handler arrays for the methods into a single exception handler table, the combining comprising positioning the exception handler arrays in a second portion of the package based on the ordering for the methods.

58. (Previously Presented) The apparatus of claim 57 including means for combining all exception handler arrays for all methods in a class in the single exception handler table.

59. (Previously Presented) The apparatus of claim 57 including means for combining all exception handler arrays for all methods in all classes in the single exception handler table.

60. (Previously Presented) The apparatus of claim 57 including means for combining all exception handler arrays for all methods in a Java™ package in the single exception handler table.

61. (Previously Presented) The apparatus of claim 57 where a method is included in a class file and the means for combining all exception handler arrays includes means for combining the exception handler arrays for all methods in a class file in the single exception handler table.

62. (Previously Presented) The apparatus of claim 57 further including means for searching the exception handler table when an exception is thrown while executing one of the methods including locating a first matching exception in the single exception handler table.

63. (Previously Presented) The apparatus of claim 62 where the means for searching includes means for retrieving in order exception handler entries from the exception handler table and means for checking the type and range of each exception handler for the first matching exception handler.

64. (Previously Presented) The apparatus of claim 63, further comprising means for stopping searching if a current exception handler does not match and is the last handler for the top most level of protected code in an associated method.

65. (Previously Presented) The apparatus of claim 57 where the class files are Java™ class files.

66. (Cancelled)

67. (Cancelled)

68. (Cancelled)

69. (Previously Presented) The apparatus of claim 57 where the machine is a virtual machine implemented on a resource constrained device.

70. (Previously Presented) The apparatus of claim 69 where the resource constrained device is a smart card.

71. (Previously Presented) The apparatus of claim 70 where the methods in one or more classes are grouped in a package and the package is installed on the smart card.

72. (Cancelled)

73. (Currently Amended) An apparatus for minimizing the amount of storage required for a runtime stack when executing a program, the runtime stack maintained at runtime during the execution of the program on a resource-constrained device for storing one or more frames where each frame includes a return pointer to an invoking method that called a currently executing method in the program, the apparatus comprising:

means for defining an ordering for methods included in the program, each method including protected code and an exception handler array defining exception handlers associated with the method, the ordering defining the placement of the protected code in a first portion of a package; and

means for combining in block form exception handler information for the methods included in the program into a combined exception handler table, the combining comprising positioning the exception handler arrays in a second portion of the package based on the ordering for methods included in the program; and

means for locating and searching the combined exception handler table when an exception is thrown during execution of one of the methods to locate the exception handler information without requiring the storage on the runtime stack of a pointer to the exception handler information.

74. (Previously Presented) The apparatus of claim 73 where the pointer is a direct pointer to the exception handler information.

75. (Previously Presented) The apparatus of claim 73 where the program is a Java™ program.

76. (Previously Presented) The apparatus of claim 75 where the machine is a virtual machine implementing a Java™ virtual machine.

77. (Previously Presented) The apparatus of claim 76 where the program includes a package of methods, the methods in one or more classes, and where the virtual machine is implemented in a resource constrained device on which the package is installed and executing.

78. (Previously Presented) The apparatus of claim 77 where the resource constrained device is a smart card.

79. (Previously Presented) The apparatus of claim 77, further including means for registering the package in a registry service at installation, the registry service maintaining a pointer and a range, the pointer indicating a location in the resource constrained device of the combined exception handler table associated with a given package, the range defining a range of addresses in the resource constrained device at which methods associated with the package are located.

80. (Previously Presented) The apparatus of claim 79 where the means for locating includes means for locating a package associated with a currently executing method

including means for comparing an address at which an exception was thrown against the range for each package registered in the registry service, the means for searching including means for searching the combined exception handler table associated with a located package.

81. (Previously Presented) An apparatus for converting class files into a converted applet for execution on a resource constrained device including:

means for receiving one or more class files, each class file including one or more methods, each method including protected code and an exception handler array defining exception handlers catchable by the method;

means for defining a data structure for storing the methods and exception handlers for the converted applet including a first and second portion;

means for defining an ordering for the methods and loading the methods according to the ordering in the second portion of the data structure; and

means for combining in block form the exception handler arrays for all methods in a single exception handler table including ordering the exception handler arrays according to the ordering defined for the methods and storing the single exception handler array in the first portion of the data structure.

82. (Currently Amended) An apparatus for managing exceptions throwable during execution of two or more methods in one or more classes by a virtual machine on a resource-constrained device, each method included in a class and including protected code and an exception handler array defining exception handlers associated with the

method, the protected code of the two or more methods stored in a first portion of a package according to an ordering, the individual exception handler arrays combined in block form and forming a single exception handler table for the two or more methods, the exception handler arrays positioned in a second portion of the package based on the ordering of the two or more methods, the apparatus comprising:

means for searching the exception handler table when an exception is thrown while executing one of the methods including locating a first matching exception in the single exception handler table.